ROOT

The ROOT session at CHEP was well attended with more attendant than sits available. ROOT has some development in the Core and I/O (support for newer C++ constructs, new interface to read TTrees, support exception and gcc 5 in interpreter), introducing of implicit parallelism of a couple of ROOT operation (TTree:GetEntry), improvements in machine learning (TMVA) and random numbers (MixMax engine) and few graphics primitives (TRatioPlot) but a significant fraction of the effort has focused on moving the documentation to Doxygen, consolidating the new CMake based build system and support for Jupyter notebooks and SWAN.

Significant progress has been made by Clang with some help from the ROOT team to support the concept of precompiled module in the clang compiler. This will later benefit ROOT to provide a precompiled version of the headers to remove the current need to parse the user header files at run-time.

ROOT starting to take advantage of improvement in C++ via new interfaces resolving outstanding challenges but in backward incompatibility way (a once every twenty years event!) while staying familiar. The goals for these new interfaces are:

- Interoperability, Simplicity, Task Parallelism ,Robustness
- Designed for change: abstraction, separation of public and internal parts
- Improve speed where possible
- Be used in experiment code developed for CERN Run 3

The javascript version of ROOT is being extended and now contains most draw options for histograms, tooltips, highlight and zooming, TTree::Draw and a Geometry viewer.

A study to improve compression is ROOT conclude that:

- LZ4 does not have significant improvement over Zlib
- Per-TTree-entry compression can dramatically accelerate random read speed for small objects although it needs more storage.
- ROOT decompresses basket in user space where random reads in memory might slow down.
- ROOT has a lot of good features: fast sequential access, sub-branches accesses, good knowledge of event structure, etc. but there is still some room to improve.

ROOT has integrated with Jupyter notebooks both with C++ and Python notebook flavours including Inline graphics, JsROOT interactive visualisation, TMVA interactive features tab completion. This is all available in the next ROOT release (6.08) and accessible online thanks to SWAN https://swan.cern.ch

ROOT offers in v6.08 important building blocks for expression of parallelism
General purpose parallel executors exploiting multiprocessing and multithreading paradigms, mechanisms for resource protection (e.g. RWSpinMutex, per thread object management) if the user wants full control over expression of parallelism (Explicit parallelism). Interfaces for achieving parallelism relying on internal features of the library (Implicit parallelism). The new developments approached with a modern view in mind: slim and intuitive programming model, compatible with STL templates, no duplication with components the STL provides.

New TMVA version released in upcoming ROOT 6.0.8; many new features: CPU and GPU-capable Deep learning, New Preprocessing Features Cross-validation, Hyperparameter-tuning, Updated Regression Loss Functions, Parallelization, Interfaces to external Machine Learning Tools, integration with Jupyter notebooks

Axel Naumann: Status and Evolution of ROOT

Vasil Vassilev: Optimizing ROOT's Performance Using C++ Modules

Philippe Canal: ROOT and new programming paradigms

Sergey Gleyser: New Machine Learning Developments in ROOT

Enric Tejedor: Expressing Parallelism in ROOT

Danilo Piparo: New ROOT Interface: Jupyter Notebooks

Marek Szuba: RootJS: Node.js Bindings for ROOT 6

Sergey Linev: JavaScript ROOT v4

Zhang Zhe: Exploring Compression Techniques for ROOT IO

Geant4

Geant4 version 10.0 was released on December 6th, 2013, introduces multi-threading capability. Since the 10.0 release, the collaboration continues its efforts to improve the performance in both physics and computing, enrich the functionalities and offer user support. Geant4MT scaled nicely with the increase number of core even on the new Intel KNL. New physics models for coming experiments, e.g. hadronic model for multi-TeV regime (for LHC run-II/III/IV), specialized EM model for noble liquid (e.g. liquid Xe) and neutrino physics model

(for intensity frontier). Next evolution is migration to the recent C++ standards, C++11/14, that enables Geant4 to better manage threads and memories used by threads. There were extension to the Geant4 analysis tools including support for MPI.

Geant4 introduces a new repository (DoSSiER) to collect and organize Geant4 validation and regression test data as well as the experimental data used for validation. A Web Application allows easy access to the data other communities have expressed interest in using the same system (GENIE, GeantV). A web service allows programmatic access to the data from user-code.

To enable the study of beam manipulation of high- and very-high-energy particle beams, Geant4 has been adapted to allow simulation of channeling; physics processes are wrapped to modify cross-section and final states taking into account channeling of particles in crystal plane such implementing a Geant4 model for the simulation of orientational effects in straight and bent crystals for high energy charged particles.

FNAl and SLAC developed a new set of software tools to allow to easily run one or multiple variants of Geant4 simulation application, to combine the simulated results and to perform collective statistical analysis of them. A first application was to explore the sensitivity of the Geant4 Bertini Cascade model to the variation of several of its key parameters.

A conversion from SOLIDWORKS cad program to GDML was started for the LUX-ZEPLIN dark matter experiment and already support many shapes including Cone, Torus and Half-ellipsoids. For the Muon g-2 experiment a converter was implemented to transform STL 3d modeling files into Geant4 tessellated solids. This scenario allows mixing shapes design in CAD and shapes implemented only in Geant4.

Note: STL (STereoLithography) is a file format native to the stereolithography CAD software created by 3D Systems.

Makoto Asai: New developments in Geant4 version 10 series

Daren Lewis Sawkey: Recent progress of Geant4 electromagnetic physics for LHC and other applications

Andrea Dotti: Software Aspects of the Geant4 Validation Repository

Andrea Dotti: Multi-threaded Geant4 on Intel Many Integrated Core architectures

Enrico Bagli: Simulation of orientational coherent effects via Geant4

Ivana Hrivnacova: Analysis Tools in Geant4 10.2

Robert Hatcher: A Software Toolkit to Study Systematic Uncertainties of the Physics Models of the Geant4 Simulation Package

Leah Welty-Rieger: Using 3D Engineering Models in a GEANT Simulation

Sunanda Banerjee: Validation of Physics Models of Geant4 using data from CMS experiment

Carl Vuosalo: A Tool to Convert CAD Models for Importation into Geant4

GeantV

GeantV has made significant progress and is receiving this week the feedback of the community on its current status and plan (see https://indico.cern.ch/event/570876/).

At Chep GeantV presented its validation methodology, a genetic algorithm to optimize the variation run-time parameters (number baskets, size of baskets, etc.) and the computing performance of the physics models.

Soon Yung Jun: Computing Performance of GeantV Physics Models

Marilena Bandieramonte: Validation of Electromagnetic Physics Models for Parallel Computing Architectures in the GeantV project

Oksana Shadura: Stochastic optimisation of GeantV code by use of genetic algorithms